

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	61	(virtual adj counter)	US-PGPUB; USPAT	OR	ON	2007/06/23 14:33
L2	37	(virtual adj time\$2) with (lag\$4 speed drift\$4)	US-PGPUB; USPAT	OR	ON	2007/06/23 15:22
L3	40	reduc\$ with migrat\$4 with (drift\$4 penal\$4)	US-PGPUB; USPAT	OR	ON	2007/06/23 15:22
L4	10	minimiz\$ with migrat\$4 with (drift\$4 penal\$4)	US-PGPUB; USPAT	OR	ON	2007/06/23 15:23
L5	5754	minimiz\$ with (drift\$4 penal\$4)	US-PGPUB; USPAT	OR	ON	2007/06/23 15:23
L6	2	(reduc\$4 minimiz\$) with (drift\$4 penal\$4) with ((virtual adj (system machine)) VM VMM)	US-PGPUB; USPAT	OR	ON	2007/06/23 15:24
L7	181	((Virtual adj machine) VM VMM) and 703/23-28.ccls.	US-PGPUB; USPAT	OR	ON	2007/06/23 15:25
L8	159	((Virtual adj machine) VM VMM) and 703/13-22.ccls.	US-PGPUB; USPAT	OR	ON	2007/06/23 15:25
L9	312	L7 L8	US-PGPUB; USPAT	OR	ON	2007/06/23 15:25
L10	8	("5103394" "5666519" "5678028" "5737579" "5761477" "5815688" "6047381").PN. OR ("6882968").URPN.	US-PGPUB; USPAT; USOCR	OR	OFF	2007/06/23 15:48
L11	4	scala\$5 with (virtual adj (time\$2))	US-PGPUB; USPAT; USOCR	OR	OFF	2007/06/23 15:58
L12	38	scala\$5 with (virtual adj (machine))	US-PGPUB; USPAT; USOCR	OR	ON	2007/06/23 15:58
L18	15	((apparent simulat\$4 virtual) adj (time clock)) with ((real wall hardware) adj (time clock)) with (lag\$4 drift\$4 behind "catch-up" delay\$4)	US-PGPUB; USPAT	OR	ON	2007/06/23 16:12
L20	182	(suspend\$4 halt\$4) with (virtual adj machine)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/23 16:23

EAST Search History

L21	19	(US-20070033589-\$ or US-20060075402-\$ or US-20050204357-\$ or US-20060130059-\$ or US-20050027500-\$).did. or (US-6550015-\$ or US-6349388-\$ or US-4814975-\$ or US-5774479-\$ or US-5488713-\$ or US-5621912-\$ or US-5095427-\$ or US-4812967-\$ or US-6882968-\$ or US-7136800-\$ or US-7146305-\$).did. or (US-6550015-\$ or US-20020056076-\$ or EP-419723-\$).did.	US-PGPUB; USPAT; DERWENT	OR	OFF	2007/06/23 16:54
L22	6	L21 and timers	US-PGPUB; USPAT; DERWENT	OR	OFF	2007/06/23 16:59
L23	735	virtual adj time	US-PGPUB; USPAT; DERWENT	OR	OFF	2007/06/23 17:01
L24	397	L23 and ((real hardware) adj time)	US-PGPUB; USPAT; DERWENT	OR	OFF	2007/06/23 17:00
L25	397	L23 and ((wall real hardware) adj time)	US-PGPUB; USPAT; DERWENT	OR	OFF	2007/06/23 17:00
L26	123	virtual adj time with (((wall real hardware) adj time))	US-PGPUB; USPAT; DERWENT	OR	OFF	2007/06/23 17:02
L27	3	L26 and ((virtual adj (machine system)) VM VMM hypervisor)	US-PGPUB; USPAT; DERWENT	OR	OFF	2007/06/23 17:26

EAST Search History

L28	47	US-5437033-\$ DID. OR US-4811276-\$ DID. OR US-5295265-\$ DID. OR US-5023771-\$ DID. OR US-5355470-\$ DID. OR US-6412035-\$ DID. OR US-4814975-\$ DID. OR US-6373846-\$ DID. OR US-5898855-\$ DID. OR US-6208661-\$ DID. OR US-6961806-\$ DID. OR US-6996748-\$ DID. OR US-7069413-\$ DID. OR US-7082598-\$ DID. OR US-7089377-\$ DID. OR US-7111086-\$ DID. OR US-7111145-\$ DID. OR US-7124327-\$ DID. OR US-7127548-\$ DID. OR US-7149843-\$ DID. OR US-7155558-\$ DID. OR US-7191440-\$ DID. OR US-20020172202-\$ DID. OR US-20060005200-\$ DID. OR US-20030217250-\$ DID. OR US-20040003323-\$ DID. OR US-20040003324-\$ DID. OR US-20040117539-\$ DID. OR US-20040123288-\$ DID. OR US-20040205203-\$ DID. OR US-20040268347-\$ DID. OR US-20050060702-\$ DID. OR US-20050060703-\$ DID. OR US-20050071840-\$ DID. OR US-20050080753-\$ DID. OR US-20050080934-\$ DID. OR US-20050080937-\$ DID. OR US-20050081199-\$ DID. OR US-20050132362-\$ DID. OR US-20050132365-\$ DID. OR US-20050216920-\$ DID. OR US-20050223377-\$ DID. OR US-20050289542-\$ DID. OR US-20060004554-\$ DID. OR US-20060004667-\$ DID. OR US-20060005003-\$ DID. OR US-20060005184-\$ DID.	US-PGPUB; USPAT	OR	OFF	2007/06/23 17:28
S1	56	virtual adj timer	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/23 16:19

EAST Search History

S2	2	"09/247,876"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/19 14:56
S3	322	703/19.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/19 15:06
S4	596	718/1.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/19 15:08
S7	18	("3881156" "4287562" "4879733" "4912734" "4926319" "4942522" "4952367" "5042005" "5089955" "5220661" "5233573" "5325341" "5363499" "5491815" "5664167" "5724399" "5740451" "5975739").PN. OR ("6550015"). URPN.	US-PGPUB; USPAT; USOCR	OR	OFF	2007/06/20 18:23
S8	510	((speed catch) adj up) accelerat\$4) with ((virtual adj machine) VMM VM)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/21 16:56
S9	63	S8 and timer	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/21 14:14

EAST Search History

S10	40	(US-20020129338-\$ or US-20020056076-\$).did. or (US-6108309-\$ or US-6522985-\$ or US-5247653-\$ or US-5550760-\$ or US-6117181-\$ or US-6725188-\$ or US-6173249-\$ or US-6134516-\$ or US-5287461-\$ or US-6792460-\$ or US-6934755-\$ or US-6240529-\$ or US-5784552-\$ or US-5937179-\$ or US-6618839-\$ or US-6901581-\$ or US-5621912-\$ or US-6795966-\$). did.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/21 14:54
S11	510	((speed catch) adj up) accelerat\$4) with ((virtual adj machine) VMM VM)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/21 17:09
S12	73	S11 and interrupt	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/21 15:07
S13	9	S10 and (VM VMM (virtual adj machine))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/21 14:54
S14	2	(catch adj up) with virtual with (timer timming)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/21 15:08
S15	2	(catch adj up) same (virtual with (timer timming))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/21 15:08

EAST Search History

S16	18	(catch adj up) and (virtual with (timer timming))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/21 15:10
S17	598	718/1.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/21 15:11
S18	71	S17 and timer	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/21 15:12
S20	23	US-20010021969-\$ DID. OR US-20010027511-\$ DID. OR US-20010027527-\$ DID. OR US-20010037450-\$ DID. OR US-20020007456-\$ DID. OR US-20020023032-\$ DID. OR US-20020147916-\$ DID. OR US-20020166061-\$ DID. OR US-20020169717-\$ DID. OR US-20030018892-\$ DID. OR US-20030074548-\$ DID. OR US-20030115453-\$ DID. OR US-20030126442-\$ DID. OR US-20030126453-\$ DID. OR US-20030159056-\$ DID. OR US-20030188179-\$ DID. OR US-20030196085-\$ DID. OR US-20040117539-\$ DID. OR US-3699532-\$ DID. OR US-3996449-\$ DID. OR US-4037214-\$ DID. OR US-4162536-\$ DID. OR US-4207609-\$ DID.	US-PGPUB; USPAT	OR	OFF	2007/06/21 17:00

EAST Search History

S21	41	US-4276594-\$ DID. OR US-4278837-\$ DID. OR US-4307447-\$ DID. OR US-4319233-\$ DID. OR US-4319323-\$ DID. OR US-4347565-\$ DID. OR US-4366537-\$ DID. OR US-4403283-\$ DID. OR US-4419724-\$ DID. OR US-4430709-\$ DID. OR US-4521852-\$ DID. OR US-4759064-\$ DID. OR US-4795893-\$ DID. OR US-4802084-\$ DID. OR US-4825052-\$ DID. OR US-4907270-\$ DID. OR US-4907272-\$ DID. OR US-4910774-\$ DID. OR US-4975836-\$ DID. OR US-5007082-\$ DID. OR US-5022077-\$ DID. OR US-5075842-\$ DID. OR US-5079737-\$ DID. OR US-5187802-\$ DID. OR US-5230069-\$ DID. OR US-5237616-\$ DID. OR US-5255379-\$ DID. OR US-5287363-\$ DID. OR US-5293424-\$ DID. OR US-5295251-\$ DID. OR US-5317705-\$ DID. OR US-5319760-\$ DID. OR US-5361375-\$ DID. OR US-5386552-\$ DID. OR US-5421006-\$ DID. OR US-5434999-\$ DID. OR US-5437033-\$ DID. OR US-5442645-\$ DID. OR US-5455909-\$ DID. OR US-5459867-\$ DID. OR US-5459869-\$ DID.	US-PGPUB; USPAT	OR	OFF	2007/06/21 17:01
-----	----	--	--------------------	----	-----	------------------

EAST Search History

S22	84	US-5473692-\$ DID. OR US-5479509-\$ DID. OR US-5504922-\$ DID. OR US-5506975-\$ DID. OR US-5511217-\$ DID. OR US-5522075-\$ DID. OR US-5528231-\$ DID. OR US-5533126-\$ DID. OR US-5555385-\$ DID. OR US-5555414-\$ DID. OR US-5560013-\$ DID. OR US-5564040-\$ DID. OR US-5566323-\$ DID. OR US-5568552-\$ DID. OR US-5574936-\$ DID. OR US-5582717-\$ DID. OR US-5604805-\$ DID. OR US-5606617-\$ DID. OR US-5615263-\$ DID. OR US-5628022-\$ DID. OR US-5633929-\$ DID. OR US-5657445-\$ DID. OR US-5668971-\$ DID. OR US-5684948-\$ DID. OR US-5706469-\$ DID. OR US-5717903-\$ DID. OR US-5720609-\$ DID. OR US-5721222-\$ DID. OR US-5729760-\$ DID. OR US-5737604-\$ DID. OR US-5737760-\$ DID. OR US-5740178-\$ DID. OR US-5752046-\$ DID. OR US-5757919-\$ DID. OR US-5764969-\$ DID. OR US-5796835-\$ DID. OR US-5796845-\$ DID. OR US-5805712-\$ DID. OR US-5809546-\$ DID. OR US-5825875-\$ DID. OR US-5825880-\$ DID. OR US-5835594-\$ DID. OR US-5852717-\$ DID. OR US-5854913-\$ DID. OR US-5867577-\$ DID. OR US-5872994-\$ DID. OR US-5890189-\$ DID. OR US-5900606-\$ DID. OR US-5901225-\$ DID. OR US-5903752-\$ DID. OR US-5919257-\$ DID. OR US-5935242-\$ DID. OR US-5935247-\$ DID. OR US-5937063-\$ DID. OR US-5944821-\$ DID. OR US-5953502-\$ DID. OR	US-PGPUB; USPAT	OR	OFF	2007/06/21 17:03
6/23/2007 5:42:49 PM C:\Documents and Settings\axaxenam\My Documents\EAST\Workspaces\10782092.wsp	US-5956408-\$ DID. OR US-5970147-\$ DID. OR US-5978475-\$ DID. OR				Page 8	

EAST Search History

S24	51	US-6192455-\$ DID. OR US-6199152-\$ DID. OR US-6205550-\$ DID. OR US-6212635-\$ DID. OR US-6222923-\$ DID. OR US-6249872-\$ DID. OR US-6252650-\$ DID. OR US-6269392-\$ DID. OR US-6272533-\$ DID. OR US-6272637-\$ DID. OR US-6275933-\$ DID. OR US-6282650-\$ DID. OR US-6282651-\$ DID. OR US-6282657-\$ DID. OR US-6292874-\$ DID. OR US-6301646-\$ DID. OR US-6308270-\$ DID. OR US-6314409-\$ DID. OR US-6321314-\$ DID. OR US-6327652-\$ DID. OR US-6330670-\$ DID. OR US-6339815-\$ DID. OR US-6339816-\$ DID. OR US-6357004-\$ DID. OR US-6363485-\$ DID. OR US-6374286-\$ DID. OR US-6374317-\$ DID. OR US-6378068-\$ DID. OR US-6378072-\$ DID. OR US-6389537-\$ DID. OR US-6397242-\$ DID. OR US-6397379-\$ DID. OR US-6412035-\$ DID. OR US-6421702-\$ DID. OR US-6435416-\$ DID. OR US-6445797-\$ DID. OR US-6463535-\$ DID. OR US-6463537-\$ DID. OR US-6499123-\$ DID. OR US-6505279-\$ DID. OR US-6507904-\$ DID. OR US-6529909-\$ DID. OR US-6535988-\$ DID. or "6557104". pn. or US-6560627-\$ DID. OR US-6609199-\$ DID. OR US-6615278-\$ DID. OR US-6633963-\$ DID. OR US-6633981-\$ DID. OR US-6651171-\$ DID. OR US-6678825-\$ DID.	US-PGPUB; USPAT	OR	OFF	2007/06/21 17:04
S25	63	(S20 or S21 or S22 or S24) and (timer VMM VM)	US-PGPUB; USPAT	OR	OFF	2007/06/21 17:08
S26	1	virtual with timer with latency\$4	US-PGPUB; USPAT	OR	OFF	2007/06/21 17:10

EAST Search History

S27	4	((increas\$4 decreas\$4) with ((virtual adj machine) VMM VM) with latenc\$4	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/21 17:11
S28	211	((virtual adj machine) VMM VM) with timer	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/22 13:56
S29	477	((virtual adj (system machine)) VMM VM) with ((catch adj up) accelerat\$4)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/22 14:21
S30	852	((virtual adj (system machine)) VMM VM) with (frequenc\$4)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/22 14:53
S31	62	S30 and timer	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/06/22 14:24
S32	254	S30 and (index\$2 counter timer)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/22 14:24
S33	70	((virtual adj (system machine)) VMM VM) with (interrupt event) with (queue)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/22 15:16

EAST Search History

S34	6953	"718".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/22 15:11
S35	1198	S34.and timer	US-PGPUB; USPAT	OR	ON	2007/06/22 15:12
S36	514	718/106.ccls.	US-PGPUB; USPAT	OR	ON	2007/06/22 15:12
S37	87	S36 and timer	US-PGPUB; USPAT	OR	ON	2007/06/22 15:12
S38	1481	((virtual adj (system machine)) VMM VM) with (restart resume start)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/22 15:18
S39	196	S38 and S34	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/22 15:22
S40	579	"703".clas. and (virtual adj2 (computer system machine))	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/22 22:13
S41	37	((virtual adj machine) VMM VM) with drift	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/23 10:54
S42	11	(timothy with Mann).in.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/23 12:16

EAST Search History

S43	44	vmware.as.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/23 12:21
S44	4	vmware.as. and timer	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/23 12:23
S45	13	(virtual adj (system machine))with (event adj queu\$5)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/06/23 12:41
S49	181	((Virtual adj machine) VM VMM) and 703/23-28.ccls.	US-PGPUB; USPAT	OR	ON	2007/06/23 12:42
S50	159	((Virtual adj machine) VM VMM) and 703/13-22.ccls.	US-PGPUB; USPAT	OR	ON	2007/06/23 15:25
S51	20	(Virtual adj (system event timer)) and 703/23-28.ccls.	US-PGPUB; USPAT	OR	ON	2007/06/23 12:43
S52	30	(Virtual adj (system event timer)) and 703/13-22.ccls.	US-PGPUB; USPAT	OR	ON	2007/06/23 12:43



Technical Resources

To view a paper directly, click the icon.

Filter by Product:	Filter by Area:	Filter by Publisher:
<input checked="" type="checkbox"/> All Products VMware Infrastructure 3 ESX Server 2.x VirtualCenter 1.x	<input checked="" type="checkbox"/> All Areas Business Continuity (Backup & HA) Desktop Manageability & Security Development & Test	<input checked="" type="checkbox"/> All Publishers VMware Network Appliance RapidApp
<i>For multiple selection, CTRL + Click</i>		
Filter Results		

Title	Publisher	Rating	Revised
Virtualizing I/O Devices on VMware Workstation's Hosted Virtual Machine Monitor	VMware		06/24/2001
Memory Resource Management in VMware ESX Server	VMware		12/08/2002
NIC Teaming IEEE 802.3ad	VMware		08/11/2003
An Analysis of Disk Performance in VMware ESX Server Virtual Machines	VMware		10/26/2003
Configuring and Installing IBM BladeCenter	VMware		03/31/2004
Hyper-Threading Support in ESX Server 2.1	VMware		04/13/2004
Configuring and Installing HP Blade Servers	VMware		04/28/2004
Login Using Windows NT	VMware		05/13/2004
Streamlining Patch Testing and Deployment	VMware		06/15/2004
Virtual to Physical Documentation and Sample Configurations	VMware		06/18/2004
Converting Image Files into Virtual Machine Disks	VMware		07/13/2004

 Building Virtual Infrastructure with VMware VirtualCenter VMware 07/29/2004

 NIC Bonding and VLANs on IBM BladeCenter VMware 08/10/2004

 Security VMware 09/24/2004

 Best Practices for ESX Server 2 VMware 11/04/2004

 Systems Management VMware 12/06/2004

 Reduce Oracle Real Application Clusters Deployment Costs and Cycle Times VMware 12/06/2004

 HCL: VMware ESX Server Supported Backup Tools VMware 12/09/2004

 VLAN Solutions VMware 12/10/2004

 Representing Physical Machines in the Virtual World VMware 12/23/2004

 Isolating Performance Problems VMware 12/23/2004

 Using esxtop to Troubleshoot Performance Problems VMware 12/23/2004

 ESX Performance Tips and Tricks VMware 02/07/2005

 Managing Remote Access VMware 02/13/2005

 Enforcing Patch Management VMware 02/13/2005

 Best Practices for Setting up VMware ACE VMware 02/13/2005

 Reference & Planning for Virtualizing Citrix VMware 02/22/2005

 Login Using Active Directory VMware 03/02/2005

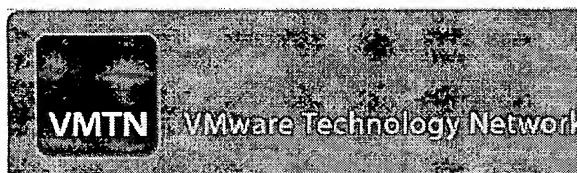
 Reference and Capacity Planning with Citrix Presentation Server (for ESX Server 2) VMware 03/09/2005

 Comparing the MUI, VirtualCenter, and vmkusage VMware 03/18/2005

1 2 3 4 5 [next page](#) [last page](#)

Copyright © 2007 VMware, Inc. All rights reserved. [Legal](#) [Privacy](#) [Accessibility](#) [Site Index](#)





Technical Resources

To view a paper directly, click the icon.

Filter by Product:

All Products

VMware Infrastructure 3

ESX Server 2.x

VirtualCenter 1.x

Filter by Area:

All Areas

Business Continuity (Backup & HA)

Desktop Manageability & Security

Development & Test

Filter by Publisher:

All Publishers

VMware

Network Appliance

RapidApp

For multiple selection, CTRL + Click

Filter Results

Title	Publisher	Rating	Revised
Virtualizing I/O Devices on VMware Workstation's Hosted Virtual Machine Monitor	VMware		06/24/2001
Memory Resource Management in VMware ESX Server	VMware		12/08/2002
NIC Teaming IEEE 802.3ad	VMware		08/11/2003
An Analysis of Disk Performance in VMware ESX Server Virtual Machines	VMware		10/26/2003
Configuring and Installing IBM BladeCenter	VMware		03/31/2004
Hyper-Threading Support in ESX Server 2.1	VMware		04/13/2004
Configuring and Installing HP Blade Servers	VMware		04/28/2004
Login Using Windows NT	VMware		05/13/2004
Streamlining Patch Testing and Deployment	VMware		06/15/2004

Virtual to Physical Documentation and Sample Configurations VMware 06/18/2004

 Converting Image Files into Virtual Machine Disks VMware 07/13/2004

 Building Virtual Infrastructure with VMware VirtualCenter VMware 07/29/2004

 NIC Bonding and VLANs on IBM BladeCenter VMware 08/10/2004

 Security VMware 09/24/2004

 Best Practices for ESX Server 2 VMware 11/04/2004

 Systems Management VMware 12/06/2004

 Reduce Oracle Real Application Clusters Deployment Costs and Cycle Times VMware 12/06/2004

 HCL: VMware ESX Server Supported Backup Tools VMware 12/09/2004

 VLAN Solutions VMware 12/10/2004

 Representing Physical Machines in the Virtual World VMware 12/23/2004

 Isolating Performance Problems VMware 12/23/2004

 Using esxtop to Troubleshoot Performance Problems VMware 12/23/2004

 ESX Performance Tips and Tricks VMware 02/07/2005

 Managing Remote Access VMware 02/13/2005

Enforcing Patch Management VMware 02/13/2005

 Best Practices for Setting up VMware ACE VMware 02/13/2005

 Reference & Planning for Virtualizing Citrix VMware 02/22/2005

 Login Using Active Directory VMware 03/02/2005

 Reference and Capacity Planning with Citrix Presentation Server (for ESX Server 2)
VMware 03/09/2005

 Comparing the MUI, VirtualCenter, and vmkusage VMware 03/18/2005

1 2 3 4 5 next page last page

Copyright © 2007 VMware, Inc. All rights reserved. [Legal](#) [Privacy](#) [Accessibility](#) [Site Index](#)



[Search Session History](#)[BROWSE](#)[SEARCH](#)[IEEE XPLORER GUIDE](#)[SUPPORT](#)

Sat, 23 Jun 2007, 12:07:06 PM EST

Search Query Display

Edit an existing query or
compose a new query in the
Search Query Display.

Select a search number (#) to:

- Add a query to the Search Query Display
- Combine search queries using AND, OR, or NOT
- Delete a search
- Run a search

	Recent Search Queries	Results
<u>#1</u>	(virtual machine<in>metadata)	2221
<u>#2</u>	((virtual machine<in>metadata))<AND>(catch-up<in>metadata))	0
<u>#3</u>	(virtual machine<in>metadata)	2221
<u>#4</u>	(virtual machine<in>metadata)	2221
<u>#5</u>	(accelerat*>speed<IN>metadata)	173380
<u>#6</u>	(virtual machine<in>metadata)	2221
<u>#7</u>	((virtual machine<in>metadata))<AND>(accelerat*>speed<in>metadata))	278
<u>#8</u>	((virtual machine<in>metadata))<AND>(accelerat*>speed<in>metadata))	278
<u>#9</u>	(virtual machine<in>metadata)	2221
<u>#10</u>	((virtual machine<in>metadata))<AND>(restart*>resum*>start*<in>metadata))	218
<u>#11</u>	(virtual machine<in>metadata)	2221
<u>#12</u>	((virtual machine<in>metadata))<AND>(suspend*<in>metadata))	5
<u>#13</u>	((virtual machine<in>metadata))<AND>(suspend*<in>metadata))	5
<u>#14</u>	((event queu*<in>metadata) <and> (virtual machine<in>metadata))	1
<u>#15</u>	((time<in>metadata) <and> (accelerat*<in>metadata))<and>(virtual machine<in>metadata)	21
<u>#16</u>	((virtual machine<in>metadata) <and> (smp<in>metadata))	9
<u>#17</u>	(virtual timer<in>metadata)	0
<u>#18</u>	(mann t.<in>au)	22
<u>#19</u>	((mann<in>au) <and>(virtual machine<in>metadata))	0
	((mann<in>au) <and>(virtual machine<in>metadata))	0

#20

<u>#21</u>	((mann<in>au))	507
<u>#22</u>	(((mann<in>au)))<AND>(virtual machine<in>metadata))	0
<u>#23</u>	((mann<in>au))	507
<u>#24</u>	((mann l.<in>au))	22
<u>#25</u>	((dependent<in>metadata) <and> (virtual machine<in>metadata))	29
<u>#26</u>	((virtual machine<in>metadata) <and> (queu* interrupt*<in>metadata))	0
<u>#27</u>	((queu* interrupt*<in>metadata))	0
<u>#28</u>	((queu* interrupt<in>metadata))	0
<u>#29</u>	((interrupt queue*<in>metadata))	0
<u>#30</u>	((interrupt queue<in>metadata))	0
<u>#31</u>	((interrupt <in>metadata))	2076
<u>#32</u>	(((interrupt <in>metadata)))<AND>(virtual machine<in>metadata))	13



[Help](#) [Contact Us](#) [Privacy & Security](#) [IEEE.org](#)
© Copyright 2006 IEEE – All Rights Reserved


[Home](#) | [Login](#) | [Logout](#) | [Access Information](#) | [Alerts](#) | [Sitemap](#) | [Help](#)

Welcome United States Patent and Trademark Office

 [Search Session History](#)[BROWSE](#)[SEARCH](#)[IEEE XPLORE GUIDE](#)[SUPPORT](#)

Sat, 23 Jun 2007, 1:41:49 PM EST

Search Query Display

Edit an existing query or
compose a new query in the
Search Query Display.

Select a search number (#) to:

- Add a query to the Search Query Display
- Combine search queries using AND, OR, or NOT
- Delete a search
- Run a search

Recent Search Queries

Results

#1 ((virtual machine<in>metadata) <and> (drift<in>metadata))

2

#2 ((virtual machine<in>metadata) <and> (suspend<in>metadata))<or> (migrat*<in>metadata)

4890

#3 ((virtual machine<in>metadata) <and> (suspend<in>metadata))<and> (resume<in>metadata)

3

#4 ((virtual machine<in>metadata) <and> (suspend*<in>metadata))<and> (resum*<in>metadata)

3

#5 (((virtual machine<in>metadata) <and> (suspend<in>metadata))<or> (migrat*<in>metadata))<AND> (timer<in>metadata))

1

[Help](#) [Contact Us](#) [Privacy & Security](#) [IEEE.org](#)

© Copyright 2006 IEEE – All Rights Reserved

Indexed by
 Inspec®

Results (page 1): #Virtual machine #time & suspend resume queue

Today's complex applications must face the distribution of data and code among different network nodes. Java is a wide-spread language that allows developers to build complex software, even distributed, but it cannot handle the migration of computations (i.e. threads), due to intrinsic limitations of many traditional JVMs. After analyzing the

Virtual Machine: exploiting some of its innovative VM techniques, ...

Keywords: Java virtual machine, code mobility, distributed applications, thread persistence

卷之三

Sched. Mining Sets And Interactive Virtual Machines Using Personal Real-time Scheduling

ACM SIGGRAPH 2005
July 31 - August 15, 2005
Proceedings of the 2005 ACM/IEEE conference on Supercomputing SCC
Volume 15
Publisher: IEEE Computer Society

Additional information: [SAS](#) [SAS/STAT](#) [SAS/IML](#) [SAS/ACROSS](#) [SAS/TEST](#)

Virtuoso must be able to mix batch and interactive VMs on the same physical hardware, thus satisfying constraint on responsiveness and compute rates for each workload. VSched is the component of Virtuoso that provides this capability. VSched is an

entirely user-level tool that interacts with the stock Linux kernel running below any type-1 virtual machine monitor to schedule VMs (Indeed, any process) ...

Foundation for an efficient multi-threaded scheme system.

January 1992 ACM SIGPLAN Lisp Pointers, Proceedings of the 1992 ACM conference
on LISP and functional programming LFP '92, Volume V Issue 1
published by ACM Press

We have built a parallel dialect of Scheme called STING that differs from its contemporaries in a number of important respects. STING is intended to be used as an operating system substrate for modern parallel programming languages. The basic

Operating system threads for modern parallel programming languages. The basic concurrency management objects in STING are first-class lightweight threads of control and virtual processors (VPs). Unlike high-level concurrency structures, STING threads and

Vps are not encumbered by complex synchronization protocols. ...

Christian D. Carlstrom, Austin McDonald, Hassan Chafi, JaeWoong Chung, Chi Cao Minh,
Christos Kozyrakis, Kunle Olukotun

June 2006 ACM SIGPLAN Notices, proceedings of the 2006 ACM SIGPLAN conference on Programming language design and implementation PLDI '06, Volume 41
Issue 6

publisher: ACM Press
full text available: <img alt="Checkmark icon" data-bbox="860 8475 875 84

Atoms is the first programming language with implicit transactions, strong atomicity, and a scalable multiprocessor implementation. Atoms is derived from Java, but replaces its

Synchronization and conditional waiting constructs with simpler transactional alternatives. The `Atoms` watch statement allows programmers to specify fine-grained watch sets used with the `Atoms` return conditional waiting statement for efficient

transactional conflict-driven wakeup even in transactional memory systems with ...

Keywords: conditional synchronization, java, multiprocessor architecture, transactional memory

DYNAMIC DYNAMICS IN COLLABORATIVE SYSTEMS

www.acm.org/results.cfm?CFID=22146450&CFTOKEN=54905001&adv=1&COLL=ACM&DL=...

- concurrency. Atomic objects have the property of running one at a time so that it can serialize the many requests sent to it.
- 11** [Xiaoli Bartham, Boris Dragovic, Kair Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Naegelbauer, Ian Pratt, Andrew Warfield](#), *Proceedings of the nineteenth ACM SIGOPS Operating Systems Review, Proceedings of the nineteenth ACM SIGOPS Operating Systems Review, Proceedings of the nineteenth ACM SIGOPS Operating Systems Review*, Volume 31 Issue 1, October 2003

Publisher: ACM Press
Full text available: [PDF \(33.63 kB\)](#) Additional Information: [bib; citation; references; citations; index](#)

The importance of supporting flexible roles and dynamic policies has long been recognized in the CSCW (Computer-Supported Cooperative Work) literature but unfortunately never with a systematic solution. This paper proposes a taxonomy of runtime dynamics in collaborative systems in general and discusses our solution in the COCA framework. Firstly, individual participants can take roles, drop roles, and switch between roles as a collaboration runs. Secondly, a role can be transferred from one ...

Keywords: CSCW, collaboration, collaborative systems, distributed computing, dynamic reconfiguration, runtime dynamics

- 8** [Process Migration](#), *ACM Computing Surveys (CSUR)*, Volume 32 Issue 3, September 2000

Publisher: ACM Press
Full text available: [PDF \(1.24 kB\)](#) Additional Information: [bib; citation; references; citations; index](#)

Process migration is the act of transferring a process between two machines. It enables dynamic load distribution, fault resilience, eased system administration, and data access locality. Despite these goals and ongoing research efforts, migration has not achieved widespread use. With the increasing deployment of distributed systems in general, and distributed operating systems in particular, process migration is again receiving more attention in both research and product development. As hi ...

Keywords: distributed operating systems, distributed systems, load distribution, process migration

- 9** [Multiple Instruction Stream Processor](#)

Richard A. Hankins, Gautham N. Chinoya, Jamison D. Collins, Perry H. Wang, Ryan Rakvic, Hong Wang, John P. Shen

- May 2006 ACM SIGARCH Computer Architecture News - Proceedings of the 33rd annual International Symposium on Computer Architecture ISCA '06**, Volume 33 Issue 2

Publisher: IEEE Computer Society, ACM Press
Full text available: [PDF \(2.03 kB\)](#) Additional Information: [bib; citation; references; index](#)

Microprocessor design is undergoing a major paradigm shift towards multi-core designs, in anticipation that future performance gains will come from exploiting thread-level parallelism in the software. To support this trend, we present a novel processor architecture called the Multiple Instruction Stream Processing (MISP) architecture. MISP introduces the sequencer as a new category of architectural resource, and defines a canonical set of instructions to support user-level inter-sequencer signal ...

- 10** [Fast References and Eviction: On Implementing Smalltalk](#)

December 1987 ACM SIGPLAN Notices - Conference proceedings on Object-oriented programming systems, languages and applications OOPSLA '87, Volume 22 Issue 12

Publisher: ACM Press
Full text available: [PDF \(22.22 kB\)](#) Additional Information: [bib; citation; references; citations; index](#)

ConcurrentSmalltalk is an object-oriented concurrent programming language/system which has been running since late 1985. ConcurrentSmalltalk has the following features: Upper-compatibility with Smalltalk-80. Asynchronous method calls and CBox objects yield

Page 3 of 6

Results (page 1): †“virtual machine” †timer suspend resume queue

15. Sustainability, performance, and leadership: Towards virtual teachiness; leveraging feedback control theory for application selection

Knowsader: [CLI](#), [checkboxes](#), [drag-and-drop](#), [real-time applications](#), [scheduleable maintenance](#)

June 2005 Proceedings of the 1st ACM/ISENIX International conference on Virtualization and Execution Environments VEE '05
Publisher: ACM Press
Full text available at: <http://www.cs.vt.edu/~mccoll/VEE05/> Additional Information: <http://www.cs.vt.edu/~mccoll/VEE05/>

19 Praktische Gestaltung: The passive programming approach with of Java/J2ME

running on the same host, it is necessary to devise techniques that enable multiple VMs to share underlying resources both fairly and efficiently. To that end, one common approach is to deploy complex resource management techniques in the hosting infrastructure. Alternatively, in this paper, we advocate the use of self-adaptation in the VMs themselves, based on feedback about resource usage and availability. Co ...

Keywords: feedback Control, friendly virtual machines, resource management

to address multiple concurrency issues in applications. However, threads are not always fine-grained enough to be successfully applied in all circumstances, especially when it

16 ASPECT-OIENTED
APPLICATIONS: ASPECT-ORIENTED
SOFTWAE ENGINEERING FOR WEB SERVERS

Kenichi Kourai, Hideaki Hibino, Shigeru Chiba
March 2007 **Proceedings of the 6th International conference on Aspect-oriented software development AOSD '07**

The Impact of Operating Strategic Business Units via Synergistic Methods
Performance of Multi-Unit Firms
Anoop Gupta, Andrew Tucker, Shigeru Ushibara
April 1991 ACM SIGMETRICS Performance Evaluation Review, Proceedings of the
1991 ACM SIGMETRICS conference on Measurement and modeling of
computer systems SIGMETRICS '91 (Volume 19, Number 1)

Full text available at www.jstor.org **Published by** ACM Press

Shared-memory multiprocessors are frequently used as compute servers with multiple parallel applications running at the same time. In such environments the efficiency of memory access is critical.

parallel application can be significantly affected by the operating system scheduling policy. In this paper, we use detailed simulation studies to evaluate the performance characteristics of parallel applications under different scheduling policies.

coscheduling or gang scheduling, process control with processor pa ...

Results 1 - 20 of 200

ପାତ୍ର କିମ୍ବା ଅନ୍ତରେ ଏହାର ପାଦରେ ଉଚ୍ଚତା କିମ୍ବା ଲାଗୁ କରିବାର ପାଇଁ ଏହାର ପାଦରେ ଉଚ୍ଚତା କିମ୍ବା ଲାଗୁ କରିବାର ପାଇଁ

Useful downloads: [Android apps](#) | [Android ROMs](#) | [Sony Ericsson themes](#) | [Sony Ericsson Ringtones](#) | [Sony Ericsson Wallpapers](#)

Search: The ACM Digital Library

Sort results by relevance expanded form Open results in a new window

Display

Results 1 - 20 of 27

Result page: 1

1 [Dynamic instrumentation of threaded applications](#)

Zhi Chen Xu, Barton P. Miller, Oscar Naini. [ACM SIGPLAN Notices, Proceedings of the seventh ACM SIGPLAN symposium on Principles and practice of parallel programming PROPP '99](#), May 1999, Volume 34 Issue 8. Publisher: ACM Press. Additional Information: 14 pages, 3 figures, 1 reference, 12 pages, 1 figure.

The use of threads is becoming commonplace in both sequential and parallel programs. This paper describes our design and initial experience with non-trace based performance instrumentation techniques for threaded programs. Our goal is to provide detailed performance data while maintaining control of instrumentation costs. We have extended Paradyne's dynamic instrumentation (which can instrument programs without recompiling or relinking) to handle threaded programs. Controlling instrumentation cost ...

2 [Characterization and computation performance of the CM-5](#)

T. T. Kwan, B. K. Totty, D. A. Reed. [1993 Proceedings of the 1993 ACM/IEEE conference on Supercomputing - Supercomputing '93](#), December 1993. Publisher: ACM Press. Full text available: [PDF](#) [301KB] [PS](#) [521KB]. Additional Information: 14 pages, 1 figure, 12 pages, 1 figure, 12 pages, 1 figure.

3 [Mechanisms for detecting and handing timer errors](#)

David B. Stewart, Pradeep K. Khosla. [January 1997 Communications of the ACM, volume 40 Issue 1](#). Publisher: ACM Press. Full text available: [PDF](#) [202KB] [PS](#) [335KB]. Additional Information: 8 pages, 1 figure, 8 pages, 1 figure, 8 pages, 1 figure.

4 [A contribution to communication systems: Organization an OBS scheduler buffer evaluation methodologies and tools valuetools '06](#)

Andrew Zalesky. [October 2006 Proceedings of the 1st international conference on Performance evaluation methodologies and tools valuetools '06](#). Publisher: ACM Press. Full text available: [PDF](#) [157KB] [PS](#) [153KB]. Additional Information: 18 pages, 1 figure, 18 pages, 1 figure.

5 [Experiment with a real-time traffic analyzer for multi-threaded applications](#)

Gilbert J. Hansen, Charles A. Linthicum, Gary Brooks. [November 1990 Proceedings of the 1990 ACM/IEEE conference on Supercomputing - Supercomputing '90](#), November 1990. Publisher: IEEE Computer Society. Full text available: [PDF](#) [771KB] [PS](#) [321KB]. Additional Information: 12 pages, 3 figures, 1 reference.

Determining the effectiveness of parallelization requires performance data about elapsed process time and total CPU time. Furthermore, it is desirable not to have to run a parallel application in a stand-alone environment in order to obtain the profile. This paper describes the CONVEY® performance analyzer, *Capa™*, with the capability to monitor parallel regions of code, in particular loops, executed in a time-sharing environment. The means by which profiling information is ...

6 [Parity and the UNIX operating system](#)

Daniel A. Canas, Laura M. Esquivel. [April 1988 ACM SIGOPS Operating Systems Review, Volume 22 Issue 2](#). Publisher: ACM Press. Full text available: [PDF](#) [13KB] [PS](#) [32KB]. Additional Information: 38 pages, 3 figures, 1 table, 32 pages.

This paper studies the compatibility degree between two different UNIX® environments. The porting of ConcurrentC (concurrent version of the C programming language) from a VAX® computer under UNIX 4.2BSD to an AT&T UNIX PC running UNIX System V was used as a testbed. ConcurrentC is analyzed to highlight the existence of the 4.2BSD features missing in UNIX System V. These characteristics are mainly related to system calls used for interrupt handling, and for process sync ...

7 [Estimating the memory overhead required for COMA architectures](#)

T. Joe, J. L. Hennessy. [April 1994 ACM SIGARCH Computer Architecture News, Proceedings of the 21ST annual International Symposium on Computer Architecture ISCA '94, Volume 22 Issue 2](#). Publisher: IEEE Computer Society Press, ACM Press. Full text available: [PDF](#) [11KB] [PS](#) [24KB]. Additional Information: 32 pages, 23 figures, 12 pages, 1 figure.

Cache-only memory architectures (COMA) have an inherent memory overhead due to the organization of main memory as a large cache called an attachment memory. This overhead consists of memory left unallocated for performance reasons as well as additional physical memory required due to the cache organization of memory. In this work, we examine the effect of data reshuffling and data replication on the memory overhead. Data reshuffling occurs when space needs to be allocated to store a remote memory ...

8 [Communication infrastructure organization of cache-coherent NUMA and COMA architectures](#)

Per Stenstrom, Truman Joe, Anoop Gupta. [April 1992 ACM SIGARCH Computer Architecture News, Proceedings of the 19th annual International Symposium on Computer Architecture ISCA '92, Volume 20 Issue 2](#). Publisher: ACM Press. Full text available: [PDF](#) [20KB] [PS](#) [22KB]. Additional Information: 20 pages, 12 figures, 16 pages, 1 figure.

Two interesting variations of large-scale shared-memory machines that have recently emerged are cache-coherent non-uniform-memory-access machines (CC-NUMA) and cache-only memory architectures (COMA). They both have distributed main memory and use directory-based cache coherence. Unlike CC-NUMA, however, COMA machines

greater number of headers arriving to find a full buffer. For ...

Keywords: all-optical switching, blocking probability, optical burst switching, scheduler, stochastic performance modeling

<http://portal.acm.org/results.cfm?coll=ACM&di=ACM&CFID=22146450&CFTOKEN=54905001> 6/23/2007

automatically migrate and replicate data at the main-memory level in cache-line sized chunks. This paper compares the performance of these two classes ...

- 9** **Processors: latency vs. energy consumption through temperature compensated lifting**
Jokim Arvidsson, Eric Park, Philip Lewis
October 2006 *Proceedings of the 4th International conference on Embedded networked sensor systems Sensys '06*
Publisher: ACM Press
Full text available: <http://portal.acm.org/cfm?coll=ACM&dl=ACM&CFID=22146450&CFTOKEN=54905001> Additional information: <http://www.csail.mit.edu/~jokim/paper.pdf>

Keywords: clock drift, sensor networks, temperature compensation, time synchronization

- 10** **The impact of operating system scheduling policies and synchronization mechanisms on the performance of distributed applications**
Anoop Gupta, Andrew Tucker, Shigeru Urushibara
April 1991 *ACM SIGMETRICS Performance Evaluation Review - Proceedings of the computer systems SIGMETRICS'91*, Volume 19 Issue 1

Publisher: ACM Press
Full text available: <http://portal.acm.org/cfm?coll=ACM&dl=ACM&CFID=22146450&CFTOKEN=54905001> Additional Information: <http://www.csail.mit.edu/~jokim/paper.pdf>

Shared-memory multiprocessors are frequently used as compute servers with multiple parallel applications executing at the same time. In such environments, the efficiency of a policy. In this paper, we use detailed simulation studies to evaluate the performance of several different scheduling strategies. These include regular priority scheduling, coscheduling or gang scheduling, process control with processor pa ...

- 11** **Virtualization and operating systems: Fuzzy techniques for managing partitioning system**
Giovanni Vassalli & Vittorio Mancuso
June 2002 *Proceedings of the 3rd International conference on Virtual execution environments VEE '02*

Publisher: ACM Press
Full text available: <http://portal.acm.org/cfm?coll=ACM&dl=ACM&CFID=22146450&CFTOKEN=54905001> Additional Information: <http://www.csail.mit.edu/~jokim/paper.pdf>

Integrating new resource management policies into operating systems (OSes) is an ongoing process. Despite innovative policy proposals being developed, it is quite difficult to deploy a new one widely because it is difficult, costly and often impractical endeavor to modify existing OSes to integrate a new policy. To address this problem, we explore the possibility of using virtual machine technology to incorporate a new policy into an existing OS without the need to make any changes to it. Thi ...

Keywords: interference, resource management, virtual machine

- 12** **Design and implementation of a framework for efficient and dynamic multi-system deployment.**
Athanasios Boulos, Chih-Chieh Han, Mani B. Srivastava
May 2003 *Proceedings of the 1st International conference on Mobile systems, applications and services MobiSys '03*

Publisher: ACM Press
Full text available: <http://portal.acm.org/cfm?coll=ACM&dl=ACM&CFID=22146450&CFTOKEN=54905001> Additional Information: <http://www.csail.mit.edu/~jokim/paper.pdf>

Wireless ad hoc sensor networks have emerged as one of the key growth areas for wireless networking and computing technologies. So far these networks/systems have been designed with static and custom architectures for specific tasks, thus providing

inflexible operation and interaction capabilities. Our vision is to create sensor networks that are open to multiple transient users with dynamic needs. Working towards this vision, we propose a framework to define and support lightweight and mobile C ...

- 13** **Virtual machines: Revit: enabling intrusion analysis through virtualization, logging and SIG-REVIEW**
George W. Dunlap, Samuel T. King, Sukru Char, Mutaz A. Basri, Peter M. Chen
December 2002 *ACM SIGOPS Operating Systems Review*, Volume 36 Issue 51
Publisher: ACM Press
Full text available: <http://portal.acm.org/cfm?coll=ACM&dl=ACM&CFID=22146450&CFTOKEN=54905001> Additional Information: <http://www.csail.mit.edu/~jokim/paper.pdf>

Current system loggers have two problems: they depend on the integrity of the operating system being logged, and they do not save sufficient information to replay and analyze attacks that include any non-deterministic events. Revit removes the dependency on the target operating system by moving it into a virtual machine and logging below the virtual machine. This allows Revit to replay the system's execution before, during, and after an intruder compromises the system, even if the intruder rep ...

- 14** **Virtual machines: Secure and performance in the Denali isolation kernel**
Andrew Whitaker, Marianne Shaw, Steven D. Grubbs
December 2002 *ACM SIGOPS Operating Systems Review*, Volume 36 Issue 51
Publisher: ACM Press
Full text available: <http://portal.acm.org/cfm?coll=ACM&dl=ACM&CFID=22146450&CFTOKEN=54905001> Additional Information: <http://www.csail.mit.edu/~jokim/paper.pdf>

This paper describes the Denali isolation kernel, an operating system architecture that safely multiplexes a large number of untrusted Internet services on shared hardware. Denali's goal is to allow new Internet services to be "pushed" into third party infrastructure, relieving Internet service authors from the burden of acquiring and maintaining physical infrastructure. Our isolation kernel exposes a virtual machine abstraction, but unlike conventional virtual machine monitors, Denali does not ...

- 15** **PL/I: Efficiency**
Michael McNeil, William Tracz
June 1980 *ACM SIGPLAN Notices*, Volume 15 Issue 6
Publisher: ACM Press
Full text available: <http://portal.acm.org/cfm?coll=ACM&dl=ACM&CFID=22146450&CFTOKEN=54905001> Additional Information: <http://www.csail.mit.edu/~jokim/paper.pdf>

All PL/I Programmers should be aware of and genuinely concerned about PL/I program efficiency. This paper addresses the following question: "How do you write a PL/I program which the PL/I Compiler will reduce to the smallest and fastest executing machine language module?" The real world payoffs of knowing how the PL/I Optimizing Compiler handles different syntactical representations of similar semantic relationships with respect to code generation and storage allocation can increase program runtime ...

- 16** **Xen: strict, strict, strict!**
Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, Andrew Waugh
October 2003 *ACM SIGOPS Operating Systems Review - Proceedings of the nineteenth ACM symposium on Operating systems principles SOSP '03*, Volume 37 Issue 5
Publisher: ACM Press
Full text available: <http://portal.acm.org/cfm?coll=ACM&dl=ACM&CFID=22146450&CFTOKEN=54905001> Additional Information: <http://www.csail.mit.edu/~jokim/paper.pdf>

Numerous systems have been designed which use virtualization to subdivide the simple resources of a modern computer. Some require special hardware, or cannot support commodity operating systems. Some target 100% binary compatibility at the expense of performance. Others sacrifice security or functionality for speed. Few offer resource isolation or performance guarantees; most provide only best-effort provisioning, risking denial of service. This paper presents Xen, an x86 virtual machine monit ...

Keywords: hypervisors, paravirtualization, virtual machine monitors

17 Process exchange on the PRIME family of computers

Edward A. Feustel

March 1984

ACM SIGARCH Computer Architecture News, volume 12 issue 1

Full text available:

[PDF]

[HTML]

[BibTeX]

[CrossRef]

[References]

[Cited by]

Search: The ACM Digital Library | The Guide

Found 931 of 204,472

Virtual time*

Terms used: virtual time

Sort results by relevance expanded form

Display results in a new window

Results page: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200

1 Virtual time
David R. Jefferson
July 1985 ACM Transactions on Programming Languages and Systems (TOPLAS), volume 7 issue 3
Publisher: ACM Press
Full text available: Additional Information: citations, references, editor's index, terms, topics

Virtual time is a new paradigm for organizing and synchronizing distributed systems which can be applied to such problems as distributed discrete event simulation and distributed database concurrency control. Virtual time provides a flexible abstraction of real time in much the same way that virtual memory provides an abstraction of real memory. It is implemented using the Time Warp mechanism, a synchronization protocol distinguished by its reliance on lookahead-rollbacks, a ...

2 The virtual time machine
R. M. Fujimoto
March 1989 Proceedings of the first annual ACM symposium on Parallel algorithms and architectures SPAA '89
Publisher: ACM Press
Full text available: Additional Information: citations, references, editor's index, terms, topics

3 Enhanced-virtual-time (EVT) scheduling: supporting heterogeneous threads in a general-purpose scheduling scheme
Kenneth J. Duda, David R. Cheriton
December 1999 ACM SIGOPS Operating Systems Review, Proceedings of the seventeenth ACM symposium on Operating systems principles SOSP '99
Publisher: ACM Press
Full text available: Additional Information: citations, references, editor's index, terms, topics

Global virtual time (GVT) is used in the Time Warp synchronization mechanism to perform irrevocable operations such as I/O and to reclaim storage. Most existing algorithms for computing GVT assume a message-passing programming model. Here, GVT computation is examined in the context of a shared-memory model. We observe that computation of GVT is much simpler in shared-memory multiprocessors because these machines normally guarantee that no two processors will observe a set of memory operations ...

4 Virtual time: storage management in conservative and optimistic systems
David Jefferson
August 1990 Proceedings of the ninth annual ACM symposium on Principles of distributed computing PODC '90
http://portal.acm.org/results.cfm?coll=ACM&di=ACM&CFID=22146450&CFTOKEN=54905001

5 An algorithm for minimally robust global virtual time
Alexander T. Tomlinson, Vijay K. Garg
July 1993 ACM SIGSIM Simulation Digest: Proceedings of the seventh workshop on Parallel and distributed simulation PADS '93, volume 23 issue 1
Publisher: ACM Press
Full text available: Additional Information: citations, references, editor's index, terms, topics

Global virtual time (GVT) is used in distributed simulations to reclaim memory, commit output, detect termination, and handle errors. It is a global function that is computed many times during the course of a simulation. A small GVT latency (delay between its occurrence and detection) allows for more efficient use of resources. We present an algorithm which minimizes the latency, and we prove its correctness. The algorithm is unique in that a target virtual time (TVT) is predetermined by an ...

6 Synchronization, global virtual time, simulation, memory, and GVT
Richard M. Fujimoto, Maria Hybinette
October 1997 ACM Transactions on Modeling and Computer Simulation (TOMACS), volume 7 issue 4
Publisher: ACM Press
Full text available: Additional Information: citations, references, editor's index, terms, topics

Global virtual time (GVT) is used in the Time Warp synchronization mechanism to perform irrevocable operations such as I/O and to reclaim storage. Most existing algorithms for computing GVT assume a message-passing programming model. Here, GVT computation is examined in the context of a shared-memory model. We observe that computation of GVT is much simpler in shared-memory multiprocessors because these machines normally guarantee that no two processors will observe a set of memory operations ...

7 Global virtual time and distributed synchronization
Jeffrey S. Steinman, Craig A. Lee, Linda F. Wilson, David M. Nicol
July 1995 ACM SIGSIM Simulation Digest: Proceedings of the ninth workshop on Parallel and distributed simulation PADS '95, volume 25 issue 1
Publisher: IEEE Computer Society, ACM Press
Full text available: Additional Information: citations, references, editor's index, terms, topics

Global Virtual Time (GVT) is the fundamental synchronization concept in optimistic simulations. It is defined as the earliest time tag within the set of unprocessed pending events in distributed simulation. A number of techniques for determining GVT have been proposed in recent years, each having their own intrinsic properties. However, most of these techniques either focus on specific types of simulation problems or assume specific hardware support. This paper specifically addresses the GVT ...

Keywords: GVT computation, SPEEDES GVT, SPEEDES framework, Synchronous Parallel Environment for Emulation and Discrete-Event Simulation framework, digital simulation, distributed simulation, distributed synchronization, efficiency, event processing, flow control, fundamental synchronization concern, global reduction operations, global virtual time, interactive support, message passing, optimistic simulations, parallel programming, portability, real time use, real-time systems, scalability, software fault tolerance, synchronisation, unprocessed pending events

8 The virtual machine
Richard M. Fujimoto
March 1991 ACM SIGARCH Computer Architecture News, volume 19 issue 1
Publisher: ACM Press

Full text available: Additional Information: citations, references, editor's index, terms, topics

5 An algorithm for minimally robust global virtual time
Alexander T. Tomlinson, Vijay K. Garg
July 1993 ACM SIGSIM Simulation Digest: Proceedings of the seventh workshop on Parallel and distributed simulation PADS '93, volume 23 issue 1
Publisher: ACM Press
Full text available: Additional Information: citations, references, editor's index, terms, topics

Global virtual time (GVT) is used in distributed simulations to reclaim memory, commit output, detect termination, and handle errors. It is a global function that is computed many times during the course of a simulation. A small GVT latency (delay between its occurrence and detection) allows for more efficient use of resources. We present an algorithm which minimizes the latency, and we prove its correctness. The algorithm is unique in that a target virtual time (TVT) is predetermined by an ...

6 Synchronization, global virtual time, simulation, memory, and GVT
Richard M. Fujimoto, Maria Hybinette
October 1997 ACM Transactions on Modeling and Computer Simulation (TOMACS), volume 7 issue 4
Publisher: ACM Press
Full text available: Additional Information: citations, references, editor's index, terms, topics

Global virtual time (GVT) is used in the Time Warp synchronization mechanism to perform irrevocable operations such as I/O and to reclaim storage. Most existing algorithms for computing GVT assume a message-passing programming model. Here, GVT computation is examined in the context of a shared-memory model. We observe that computation of GVT is much simpler in shared-memory multiprocessors because these machines normally guarantee that no two processors will observe a set of memory operations ...

7 Global virtual time and distributed synchronization
Jeffrey S. Steinman, Craig A. Lee, Linda F. Wilson, David M. Nicol
July 1995 ACM SIGSIM Simulation Digest: Proceedings of the ninth workshop on Parallel and distributed simulation PADS '95, volume 25 issue 1
Publisher: IEEE Computer Society, ACM Press
Full text available: Additional Information: citations, references, editor's index, terms, topics

Global Virtual Time (GVT) is the fundamental synchronization concept in optimistic simulations. It is defined as the earliest time tag within the set of unprocessed pending events in distributed simulation. A number of techniques for determining GVT have been proposed in recent years, each having their own intrinsic properties. However, most of these techniques either focus on specific types of simulation problems or assume specific hardware support. This paper specifically addresses the GVT ...

Keywords: GVT computation, SPEEDES GVT, SPEEDES framework, Synchronous Parallel Environment for Emulation and Discrete-Event Simulation framework, digital simulation, distributed simulation, distributed synchronization, efficiency, event processing, flow control, fundamental synchronization concern, global reduction operations, global virtual time, interactive support, message passing, optimistic simulations, parallel programming, portability, real time use, real-time systems, scalability, software fault tolerance, synchronisation, unprocessed pending events

8 The virtual machine
Richard M. Fujimoto
March 1991 ACM SIGARCH Computer Architecture News, volume 19 issue 1
Publisher: ACM Press

http://portal.acm.org/results.cfm?coll=ACM&di=ACM&CFID=22146450&CFTOKEN=54905001

6/23/2007

- 9** **Source-level debugging of automatically parallelized code**
Robert Corn
December 1991 **ACM SIGPLAN Notices , Proceedings of the 1991 ACM/ONR workshop on Parallel and distributed debugging PADD '91**, volume 26 issue 12
Full text available: [http://portal.acm.org/.../coll=ACM&di=ACM&CFID=22146450&CFTOKEN=54905001](#) Additional information: [http://portal.acm.org/.../coll=ACM&di=ACM&CFID=22146450&CFTOKEN=54905001](#)
- 10** **Bounds and approximations for self-initiating distributed simulation without lookahead**
Robert E. Felderman, Leonard Kleinrock
October 1991 **ACM Transactions on Modeling and Computer Simulation (TOMACS)**, Volume 1 Issue 4
Publisher: ACM Press
Full text available: [http://portal.acm.org/.../coll=ACM&di=ACM&CFID=22146450&CFTOKEN=54905001](#) Additional information: [http://portal.acm.org/.../coll=ACM&di=ACM&CFID=22146450&CFTOKEN=54905001](#)
- We provide upper and lower bounds and an approximation for speedup of an optimistic self-initiated distributed simulation using a very simple model. We assume an arbitrary number of processors and a uniform connection topology. By showing that the lower bound increases essentially linearly with P , the number of processors, we find that the optimistic approach scales well as P increases. The model tracks the progress of Global Virtual Time (GVT) and eliminates ...
- Keywords:** Global Virtual Time, Time Warp, bounds, discrete-event simulation, performance analysis, speedup, virtual time
- 11** **Time warp operating system**
D. Jefferson, B. Beckman, F. Wieland, L. Blume, M. Diloreto
November 1987 **ACM SIGOPS Operating Systems Review , Proceedings of the eleventh ACM Symposium on Operating systems principles SOSP '87**, volume 21 issue 5
Publisher: ACM Press
Full text available: [http://portal.acm.org/.../coll=ACM&di=ACM&CFID=22146450&CFTOKEN=54905001](#) Additional information: [http://portal.acm.org/.../coll=ACM&di=ACM&CFID=22146450&CFTOKEN=54905001](#)
- This paper describes the Time Warp Operating System, under development for three years at the Jet Propulsion Laboratory for the Caltech Mark III Hypercube multi-processor. Its primary goal is concurrent execution of large, irregular discrete event simulations at maximum speed. It also supports any other distributed applications that are synchronized by virtual time. The Time Warp Operating System includes a complete implementation of the Time Warp mechanism, and is a substantial d ...
- 12** **A hierarchical fair service discipline for link-sharing, best-effort, and priority**
Ion Stoica, Hui Zhang, T. S. Eugene Ng
April 2000 **IEEE/ACM Transactions on Networking (TON)**, volume 8 issue 2
Full text available: [http://portal.acm.org/.../coll=ACM&di=ACM&CFID=22146450&CFTOKEN=54905001](#) Additional information: [http://portal.acm.org/.../coll=ACM&di=ACM&CFID=22146450&CFTOKEN=54905001](#)
- In this paper, we study hierarchical resource management models and algorithms that support both link-sharing and guaranteed real-time services with decoupled delay (priority) and bandwidth allocation. We extend the service curve based QoS model, which defines both delay and bandwidth requirements of a class, to include fairness, which is important for the integration of real-time and hierarchical link-sharing services. The resulting Fair Service Curve /link-sharing model formalizes the go ...
- 13** **A hierarchical fair service curve architecture for link-sharing, real-time and priority**
Ion Stoica, Hui Zhang, T. S. Eugene Ng
October 1997 **ACM SIGCOMM Computer Communication Review , Proceedings of the ACM SIGCOMM '97 conference on Applications, technologies, architectures, and protocols for computer communications SIGCOMM '97**, volume 27 issue 4
Publisher: ACM Press
Full text available: [http://portal.acm.org/.../coll=ACM&di=ACM&CFID=22146450&CFTOKEN=54905001](#) Additional information: [http://portal.acm.org/.../coll=ACM&di=ACM&CFID=22146450&CFTOKEN=54905001](#)
- In this paper, we study hierarchical resource management models and algorithms that support both link-sharing and guaranteed real-time services with decoupled delay (priority) and bandwidth allocation. We extend the service curve based QoS model, which defines both delay and bandwidth requirements of a class, to include fairness, which is important for the integration of real-time and hierarchical link-sharing services. The resulting Fair Service Curve /link-sharing model formalizes the go ...
- 14** **Hierarchical packet fair reservation scheduling**
Jon C. R. Bennett, Hui Zhang
October 1997 **IEEE/ACM Transactions on Networking (TON)**, volume 5 issue 5
Publisher: IEEE Press
Full text available: [http://portal.acm.org/.../coll=ACM&di=ACM&CFID=22146450&CFTOKEN=54905001](#) Additional information: [http://portal.acm.org/.../coll=ACM&di=ACM&CFID=22146450&CFTOKEN=54905001](#)
- For scalable support of guaranteed services that decouples the QoS control plane from the packet forwarding plane. More specifically, under this architecture, core routers do not maintain any QoS reservation states, whether per-flow or aggregate. Instead, QoS reservation states are stored at and managed by bandwidth broker(s). There are several advantages of such a bandwidth broker architecture. Among others, it relieves core routers of QoS control functions such as administ ...
- 15** **Decoupling QoS control from core routers: a new bandwidth broker architecture for scalable support of guaranteed services**
Zhi-Li Zhang, Zheng Duan, Linlin Gao, Yiwel Thomas Hou
August 2000 **ACM SIGCOMM Computer Communication Review , Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication SIGCOMM '00**, volume 30 issue 4
Publisher: ACM Press
Full text available: [http://portal.acm.org/.../coll=ACM&di=ACM&CFID=22146450&CFTOKEN=54905001](#) Additional information: [http://portal.acm.org/.../coll=ACM&di=ACM&CFID=22146450&CFTOKEN=54905001](#)
- Although users may want to employ shared variables when they program distributed simulation applications, almost none of the currently existing distributed simulation systems does offer this facility. In this paper, we systematically present new algorithms which provide consistent shared variables for distributed simulation applications. Basically, our approach combines known techniques to isolate distributed shared memory with simulation algorithms. As there are essentially two classes of distri ...
- 16** **How to interface shared variables in distributed simulation**
Horst Mehl, Stefan Hammes
September 1995 **ACM SIGSIM Simulation Digest**, Volume 23 issue 2
Publisher: ACM Press
Full text available: [http://portal.acm.org/.../coll=ACM&di=ACM&CFID=22146450&CFTOKEN=54905001](#) Additional information: [http://portal.acm.org/.../coll=ACM&di=ACM&CFID=22146450&CFTOKEN=54905001](#)
- Keywords:** fairness, link-sharing, packet scheduling, quality of service (QoS), real-time

Keywords: conservative simulation, discrete event simulation, distributed simulation, optimistic simulation, shared variables, distributed shared memory, distributed simulation, optimistic simulation, shared variables, fault tolerance.

- 17 **Exact GPS scheduling with lookahead, synchronicity, and its application to real-time optimality**
Pablo Valente

- August 2004 **ACM SIGCOMM Computer Communication Review, Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications SIGCOMM '04**, volume 34 Issue 4

- Publisher: ACM Press
Full text available: Additional Information:

Generalized Processor Sharing (GPS) is a fluid scheduling policy providing perfect fairness. The minimum deviation (lead/lag) with respect to the GPS service achievable by a packet scheduler is one packet size. To the best of our knowledge, the only packet scheduler guaranteeing such minimum deviation is Worst-case Fair Weighted Fair Queuing (WFQ), that requires on-line GPS simulation. Existing algorithms to perform GPS simulation have O(M) complexity per packet transit ...

Keywords: computational complexity, data structures, packet scheduling, quality of service

- 18 **Optimistic distributed simulation based on transitive dependency tracking**

- June 1997 **ACM SIGSIM Simulation Digest, Proceedings of the eleventh workshop on Parallel and distributed simulation PADIS '97**, Volume 27 Issue 1

- Publisher: IEEE Computer Society, ACM Press
Full text available: Additional Information:

In traditional optimistic distributed simulation protocols, a logical process (LP) receiving a straggler rolls back and sends out anti-messages. The receiver of an anti-message may also roll back and send out more anti-messages. So a single straggler may result in a large number of anti-messages and multiple rollbacks of some LPs. In the authors' protocol, an LP receiving a straggler broadcasts its rollback. On receiving this announcement, other LPs may roll back but they do not announce their ...

Keywords: anti-messages, dependency information, distributed recovery, logical process, memory management, message tagging, optimistic distributed simulation, optimistic distributed simulation protocols, process rollback, rollback broadcasting, straggler, time warp simulation, transitive dependency information, transitive dependency tracking

- 19 **A virtual machine performance analysis**
M. D. Canon, D. H. Fritz, J. H. Howard, T. D. Howell, M. F. Mittona, J. Rodriguez-Rosell

- February 1990 **Communications of the ACM**, Volume 23 Issue 2

- Publisher: ACM Press
Full text available: Additional Information:

Keywords: computer system simulation, performance evaluation, virtual machines

- 20 **Performance analysis of time warp with hypercube multiprocessors and exponential back-off times**

- Anup Gupta, Ian Akyildiz, Richard M. Fujimoto
April 1991 **ACM SIGMETRICS Performance Evaluation Review, Proceedings of the 1991 ACM SIGMETRICS conference on Measurement and modeling of**